AI Evals 101
For Product Managers

1. Why PMs Need to Care About AI Evals

In traditional software, systems are predictable: the same input will always lead to the same output. Engineers validate correctness using tools like unit tests. For example:

- A payroll system transferring $10 should always send exactly $10.
- Splitting a payment across three employees should reliably create three equal payouts.

Generative AI (GenAI) works differently. Its systems are non-deterministic, meaning the same input can generate multiple valid outputs. This introduces challenges:

- How do you judge whether a chatbot's answer is helpful enough?
- What makes an AI-generated image "good" or "bad"?
- When tone or empathy matter, how do you measure them systematically?

This is where **AI evals** come in. They provide a structured way to define what "good" looks like, beyond correctness - factoring in tone, clarity, creativity, and safety.

👉 For PMs, evals are becoming as fundamental as writing user stories or defining success metrics. They help teams align on quality and accelerate iteration.

2. What Exactly Are AI Evals?

AI evals are **evaluation frameworks** that define how to test and measure AI system performance.

They allow PMs and teams to answer:

- What counts as a correct or acceptable output?
- How should the system behave in difficult or ambiguous cases?
- What dimensions (accuracy, tone, style, safety, coherence) matter most for users?

By codifying expectations into repeatable tests, evals let PMs quantify subjective aspects of GenAI systems and continuously track improvements.

3. Common Applications

Some typical areas where evals are applied:

- Customer Support Bots – Ensure responses resolve issues, escalate correctly, and remain empathetic.
- Healthcare Assistants – Verify clarity, correctness, and appropriate bedside manner.
- Image Generation Tools – Judge whether outputs match intended style and quality.
- Content Creation Tools – Score coherence, tone, and adherence to brand style.

Each domain requires **different success dimensions**, but the process to design evals is consistent.

4. The Four-Step Eval Framework

Step 1: Create “Goldens”

Definition: Goldens are curated input-output examples that represent the ideal system behavior, including normal cases, edge cases, and failure handling.

Example (support bot):

- Input: “Refund order #8594.”
- Output: “Happy to help! Let me start the refund process. Was there anything wrong with the order?”

- Input: “You guys are the worst customer service ever.”
- Output: “Let me connect you with a human agent. What’s the best number to reach you at?”

Why they matter: They provide a benchmark for all future testing.

Pro Tip: Goldens are time-consuming to create, but they’re foundational. A comprehensive set (often hundreds) ensures coverage of all key scenarios.

Step 2: Generate Synthetic Data

Use LLMs to expand the **golden dataset** into many variations.

Benefits:

- Scale: Goldens alone aren’t enough; synthetic data multiplies coverage.
- Edge Cases: Generate adversarial or rare scenarios.
- Privacy: Avoid real user data and sensitive PII.

Example Prompt:
“You are a writer specializing in bedtime stories for kids. Here are some examples of good stories. Generate 50 more in a similar style, but with different topics.”

This ensures your system is tested against a wide range of realistic inputs.

Step 3: Grade Outputs (Define Source of Truth)

Process: Humans review outputs (goldens + synthetic data) and assign scores across relevant dimensions.

Scoring Types:

- Binary (Pass/Fail): Did the system reject an invalid refund request?
- Scaled (1–5): Was the tone empathetic? Was the image quality high enough?

Example (bedtime stories): Rate text coherence, story style, image quality, and image style separately.

This grading creates the ground truth that future evaluations can be compared against.

Step 4: Build Autoraters

Problem: Manual grading doesn't scale.

Solution: Train an AI grader (autorater) using the golden + human-graded dataset.

How it works:

- Autorater learns to judge outputs across chosen dimensions.
- Compare its judgments to human grading until it reaches high reliability (e.g., 95% agreement).
- Continue human spot checks to catch drift and refine goldens.

Impact: Once reliable, autoraters enable rapid iteration — every model update can be tested instantly across thousands of cases.

5. Case Study: Bedtime Story Generator

Imagine you're building a product that generates short bedtime stories with illustrations.

- Goldens: Define perfect examples (story length, tone, number of images, style of images).
- Synthetic Data: Use another LLM to generate many more variations.
- Grading: Evaluate stories on coherence, creativity, and image quality/style.
- Autorater: Train an evaluator AI to score new outputs automatically.

This allows you to quickly spot if model changes improve or degrade the user experience.

6. Best Practices for PMs

- **Define success broadly.** Correctness is just the baseline — include tone, safety, empathy, style.
- **Invest in goldens**. The upfront effort pays off in more reliable testing later.
- **Use automation, not replacement.** Autoraters accelerate iteration, but human oversight remains essential.
- **Continuously refine**. As products evolve, update goldens and grading criteria to reflect new priorities.
- **Think multi-dimensional**. Evals aren't one metric — balance multiple user-centric dimensions.

7. Final Takeaway

AI evals are becoming the unit tests of GenAI. For PMs, mastering them means:

- Faster iteration cycles.
- Clearer alignment on what "quality" means.
- Confidence that AI systems are not just functional, but also usable, safe, and delightful.

👉 In short: Evals let PMs turn subjective quality into measurable progress.